HopDock: A probabilistic search algorithm for decoy sampling in protein-protein docking

Irina Hashmi[1] and Amarda Shehu[12]

[1]Department of Computer Science
[2]Department of Bioinformatics
George Mason University, 4400 University Dr., Fairfax, VA, 22030, United States

Email: Irina Hashmi - ihashmi@gmu.edu; Amarda Shehu*- amarda@gmu.edu;

*Corresponding author

Abstract

Background: Elucidating the three-dimensional structure of a higher-order molecular assembly formed by interacting molecular units, a problem commonly known as docking, is central to understanding the molecular basis of biological function in the living and diseased cell. Though protein assemblies are ubiquitous in the cell, it is currently challenging to predict the native structure of a protein assembly in silico.

Methods: This work proposes a novel probabilistic search algorithm, HopDock, to efficiently search the interaction space of protein dimers. The goal is to obtain an ensemble of low-energy dimeric configurations, also known as decoys, that can be effectively used by ab-initio docking protocols. HopDock is based on the Basin Hopping (BH) framework and repeatedly follows up a structural perturbation of a dimeric configuration with an energy minimization to explicitly sample configurations that represent local minima of a chosen energy function. HopDock employs both geometry and evolutionary conservation analysis to narrow down the interaction search space of interest for the purpose of efficiently obtaining a diverse decoy ensemble.

Results and conclusions: A detailed analysis and a comparative study on seventeen different dimers shows HopDock obtains a broad view of the energy surface near the native dimeric structure and samples many near-native configurations. The results show that HopDock has high sampling capability and can be employed to effectively obtain a large and diverse ensemble of decoy configurations that can then be further refined in greater structural detail in ab-initio docking protocols.

Background

Proteins do not operate in isolation. They achieve their biological function by interacting with each other and other molecules to form higher-order molecular assemblies. Structural characterization of protein assemblies (assemblies formed by interacting protein units) is central to understanding molecular interactions, designing new effective drugs, and elucidating the molecular basis for different biological functions in the living and diseased cells [1].

There are mainly two predominant experimental techniques to elucidate the biologically-active structure of a protein assembly: X-ray Crystallography and Nuclear Magnetic Resonance (NMR). These techniques are time- and labor-intensive and are often limited by the size of the molecular assembly [2]. The number of protein-protein assemblies with strucures deposited in the Protein Data Bank (PDB) [3] is small compared to that of single protein chains. Due to the biological importance and ubiquity of protein-protein assemblies and current limitations of experimental techniques, computational approaches are emerging to complement wet laboratory efforts in elucidating structures of protein assemblies.

When the number of protein units is limited to two, the problem of predicting the biologically-active or native structure formed upon docking of the protein units onto each other is known as protein docking. This problem is challenging to address in-silico for several reasons. Figure 1 illustrates the docking problem where two unbound units A and B interacts with each other to form a bound configuration. If no a priori information is available, then the problem requires searching over a space of $N * M + 6$ dimension, where N and M are the number of parameters to represent the unbound protein molecules, and 6 is the number of parameters to represent the spatial arrangement of one protein unit on top of another. Due to the high dimensionality of this search space, many computational approaches focus on rigid-body docking, where the monomers are considered rigid (they do not change their structure after docking). The goal is to reduce the parameter search space to SE(3), the space of spatial arrangements of one unit on the other.

Nowadays, many protein-protein docking software and web servers are available, such as pyDock [4], Haddock [2], Zdock [5], ClusPro [6], PatchDock and SymmDock [7], Combdock [8], Budda [9] ,RosettaDock [10], SKE-DOCK [11], FiberDock [12], and more. While their accuracy is steadily increasingly [13,14], summaries in the Critical Assessment of PRedicted Interactions(CAPRI) setting show that no single method is currently sufficient to successfully predict native or near-native structures for every target protein assembly [15]. Limited sampling capability, inaccuracy of the energy function used to rank an interaction interface, or a combination of both are often cited as possible reasons for current limitations.

Most docking protocols that do not employ any a priori knowledge about the location of the actual interaction interface in the native dimeric structure follow a similar template that consists of two main stages [16]. In stage one, a large ensemble of dimeric configurations is obtained. Scoring functions are used to increase the likelihood that this ensemble contains configurations near the native structure. The ensemble is reduced in preparation for stage two through either scoring functions that employ more detail or through clustering-based techniques that select a subset of the decoys generated in stage one. The selected decoys are possibly added more structural detail, refined at length through more computationally-intensive energy minimization techniques to make final predictions on which decoys best represent the sought native structure. In some protocols, flexibility is considered to improve the quality of the selected decoys and possibly get closer to the native structure [2]. An important component of the success of this two-stage protocol is the ability of the search algorithm employed in stage one to obtain a relevant ensemble of decoys and not miss the region near the native structure.

The focus of this work is on enhancing sampling of relevant regions in the dimeric configuration space to obtain a diverse decoy of ensemble that can then be analyzed and further refined in energetic detail in the context of ab-initio docking protocols. Towards this goal, we propose here a novel probabilistic search algorithm, HopDock. HopDock samples configurations that are minima of a given energy function. Its search space is SE(3), consisting of rigid-body transformations that place the given unbound structure of a protein unit, A, on top of the unbound structure of the other protein unit, B (this is illustrated in Figure 1.) The search space explored by HopDock is a subspace of SE(3) that satisfies certain constraints. This subspace consists of transformations that align regions on molecular surfaces deemed more likely to correspond to the actual interaction interfaces. Two criteria are used as constraints, geometric complementarity of the aligned regions and evolutionary conservation of surface amino acids on the regions considered for alignment.

HopDock is an evolutionary search algorithm and realizes in particular the Basin Hopping (BH) framework [17] (hence the name HopDock). Our adaptation of this framework in HopDock focuses on efficiently navigating the reduced search space to obtain an ensemble of bound configurations corresponding to local minima of a given energy function. Since the focus in this work is on proposal and analysis of effective components in the BH framework for protein docking, the energy function considered here is a simple one consisting of basic terms. Our inspiration to build over the BH framework comes from recent findings in the computational structural biology community. Though the BH framework was first proposed to compute local minima of small atomic clusters involving Lennard-jones potential, it has now been shown promising in obtaining low-energy decoy configurations in the context of ab-initio protein structure prediction, where the goal is to predict the structure of a single protein chain in isolation [18–21]. At the core of the BH framework lies a repeated application of a perturbation followed by an energy minimization to obtain a trajectory of low-energy local minima. A Metropolis criterion [22] allows biasing the trajectory towards lower-energy region of the energy surface over time. Our adaptation of the BH framework in HopDock focuses on effective implementations of the perturbation and minimization components that make use of the underlying SE(3) search space. For example, the structural perturbation in HopDock builds over the basic process of aligning geometrically-complementary and evolutionary-conserved regions on the molecular surfaces. The minimization component uses the simple energetic scheme to further optimize a configuration resulting from the structural perturbation.

A detailed analysis over different implementations of the perturbation and minimization components is carried out in this paper to obtain effective implementations of these components in HopDock. HopDock is benchmarked on a diverse list of protein dimers with known native structures. A comparative analysis places HopDock in the context of other search algorithms used in docking protocols. Our results suggest that HopDock is efficient, competitive, and samples many near-native configurations. These characteristics make it a promising search algorithm to use in the context of docking protocols, particularly if more powerful energy functions are used and if the generated decoys are further selected and refined at greater detail and with more computational resources [23].

The rest of this article is organized as follows. We first provide a review of related work in order to place HopDock in context. Details on the different components of HopDock are provided in the Methods section. The Results section evaluates these components on seventeen diverse protein dimers and further compares the result of HopDock to those reported or obtained by current state-of-the-art docking protocols. The Conclusions section provides a discussion and offers promising directions of future research.

Related work

Current docking methods can primarily be categorized into two approaches, energy-based and geometry-based. Methods like pyDock [4], RossettaDock [10], ClusPro [6], and Haddock [2] take an energy optimization approach. The optimization seeks minima of a defined energy function. If the energy function is sufficiently accurate, near-native configurations will be found among the lowest-energy minima [6, 15]. In docking protocols, the process is usually split into two stages. In the first stage, a search is conducted to obtain a large number of low-energy bound configurations. The focus on the size of the ensemble is partially due to the fact that current energy functions are not accurate. Indeed, if only the lowest-energy minimum is maintained in the ensemble, the native structure will certainly be missed by many Ångströms (described in

Results section). The size of the ensemble makes it impractical to employ a lot of structural detail and use expensive energy functions. For this reason, typically, the large ensemble is obtained with a simple scoring function. The ensemble is reduced through selection techniques, often relying on structural clustering, to obtain a subset that can be afforded to be optimized in greater structural detail and with more expensive scoring functions in stage two. Computational time can even be devoted in this stage to incorporate some flexibility around detected interfaces in the bound configurations [12].

RossettaDock is a representative of current protocols. The optimization in RosettaDock is carried out over rigid-body orientations and side-chains, followed by continuous minimization. pyDock is another optimization-based server for accurately scoring rigid-body motions. In the first stage, pyDock uses FT-DOCK [24], a Fast Fourier Transform-based docking algorithm, for rigid-body docking. Configurations are then evaluated by their binding energy based on electrostatistics and desolvation to obtain a relevant subensemble. As in pyDock, the first stage in ClusPro is performed using a Fast Fourier Transform-based docking algorithm known as DOT [25]. In preparation for stage two, configurations are filtered using a combination of desolvation and electrostatic energies. A clustering algorithm is applied to discriminate against false positives and reduce the set of configurations to near-native structures. Haddock [2] is another example of an energy-based docking protocol that makes use of biochemical data available from NMR to reduce the search space where possible.

Even if computational resources are considered unlimited for optimization, research shows that designing accurate energy functions to capture native interactions remains challenging [15, 26]. Designing energy functions to capture the true interaction interface is still desirable. For this reason, a group of docking methods take a complementary approach that delays energy considerations as much as possible.

Instead of conducting the search over a large continuous space, some methods like Budda [9], CombDock [8], PatchDock, SymmDock [7], ZDOCK [5], and LZerD [27] discretize the space by defining geometrically-complementary regions on the molecular surfaces of the unit structures to be docked. The process of searching for arrangements that take one unit over the other then becomes searching for rigid-body transformations that align a region of one molecular surface with a complementary region of the other molecular surface. The main basis of this geometric treatment is that molecules are more likely to interact along geometrically-complementary regions on their surfaces. Convex regions fit better in concave ones, which should produce more stability for docked configurations that superimpose geometrically-complementary regions.

In order to model geometric complementarity, the molecular surfaces of the unbound units need to be analyzed and summarized in terms of geometric properties. Several numerical methods quantize and represent molecular surface with a collection of points, most notably the Connolly [28] and Shuo methods [29]. These methods summarize a molecular surface in terms of "critical" points that contain information on whether the surface region they represent is convex, concave, or saddle. This information is used to consider only rigid-body transformations that align geometrically-complementary regions (such as convex with concave).

The search for geometrically-complementary surface regions is conducted through mainly two approaches. A traditional grid-based shape complementary approach like FTDock [24] identifies grid points that surround the receptor, and the total number of grid points that overlap any grid points corresponding to ligand points. A more accurate and detailed computer vision-based technique known as Geometric Hashing [30] uses transformation-invariant representations of the molecular surface which allow direct matching. It takes as input a database of objects and a scene in which to find the objects. The algorithm consists of a preprocessing stage and a recognition stage. For the case of protein docking, during the preprocessing step, some feature of the base unit is extracted and hashed into a table. The recognition step similarly extracts related features from the moving molecule and then matches those features to those of the base molecule stored in the hash table.

CombDock [8] is based on Geometric Hashing to match geometrically-complementary surface regions in the first stage and in the filtering stage it uses both geometric and physico-chemical features to identify promising decoys without any use of energy function. LZerD [27] also uses Geometric Hashing for shape matching in the first step and incorporates a novel geometry-based scoring function using 3D Zernike descriptors in the final step. Multi-LZerD, a recent algorithm for protein assemblies of more than two units [31] uses LZerD for pairwise docking and then relies on a genetic algorithm to sample multimeric configurations. The multimeric decoys are ranked with a physics-based scoring function. Other techniques, like VASP [32], perform volumetric analysis to represent protein shape and the shape of the surface cavities, clefts, and tunnels.

Due to the implicit discretization of the search space, geometry-based approaches are more efficient but also less accurate than energy-based approaches. Thus, geometry-based approaches are useful to obtain many decoys in an efficient manner. Optimization can be delegated to subsequent stages. Indeed, since their introduction, they have demonstrated that they feasibly produce decoy configurations that then, through further energetic refinement, reproduce biologically-relevant native assemblies [33].

Methods

This section provides the basis of the novel search algorithm we propose in this work. HopDock is based on the Basin Hopping framework, and the search in it is guided by a simple energy function. The search is conducted over rigid-body transformations that align interfaces that are both geometrically-complementary and evolutionary-conserved. Though these two criteria do not guarantee finding the native interaction interface, they do allow narrowing the search for dimeric configurations to those that align credible interfaces. Geometric complementarity is a well-established predictor for true contact interfaces. Moreover, a detailed analysis in the Results section shows that focusing on evolutionary-conserved regions not only helps finding the correct interaction interface, but more evolutionary-conserved regions are on the native interaction interface than elsewhere on the molecular surface.

Details of the proposed algorithm are presented as follows. First, we define the search space by describing in detail how we use the geometry and evolutionary conservation information to detect rigid-body transformations of interest. Second, we describe a simple energy function that can rank a bound configuration resulting from applications of such a rigid-body transformation. Third, we relate details on how all these elements are incorporated in the proposed HopDock algorithm.

From molecular surfaces to rigid-body transformations

We now describe how molecular surfaces are analyzed and represented in order to define rigid-body transformations that align chosen surface regions of the units being docked onto each other. Our criteria for choosing certain surface regions are geometric complementary and evolutionary conservation, as detailed below.

From molecular surfaces to critical points

The predominant representation of a molecular surface is the Connolly Surface representation [28]. The Connolly method places a probe ball, representing the solvent molecule, tangent to the atoms of the molecule on thousand different locations. For each position of the ball, the point that does not overlap with the van der Waals radii and points facing the inward-surface of the probe becomes part of the molecular surface. For each surface point, the Connolly representation maintains the 3D coordinate, the normal mode, and a numerical value to indicate the type of the point. The type ranges from convex, to saddle, to concave, depending on the tangency of the probe to the number of atoms of the molecular surface.

The Connolly representation is dense. A sparse representation that simplify the Connolly surface can be calculated as described in [29]. This representation consists of a series of critical points. Critical points are defined as the maxima or minima of a Connolly face of a molecular surface. Critical points are termed as “caps“, “pits“ and “belts“ to represent the center of gravity of the convex, concave, and saddle surface of the Connolly representation, respectively. The collection of critical points is sufficient and complete to cover key locations of the molecular surface. This sparse representation reduces the total number of points from the high number of points generated by the Connolly surface and so reduce the costs of a geometric treatment in rigid-body docking.

From critical points to active critical points: an evolutionary conservation analysis

We now introduce the notion of an active critical point by additionally considering an evolutionary conservation analysis of the molecular surface.

Several studies have shown that molecular regions that are part of interaction interfaces are under higher evolutionary pressure to maintain their functional integrity [34]. Some amino acids are bound to remain more conserved throughout evolution than others if they are involved in an interaction interface. Thus, evolutionary conservation can be a good predictor of the native interaction interface. Several methods [35], [36] now exist for rigorous evolutionary analysis of protein sequences that allow associating evolutionary conservation values with each amino acid of a protein of interest.

The evolutionary analysis method known as Joint Evolutionary Trace (JET) [35], which we employ in this work, allows associating conservation scores with each amino acid of a protein chain. JET relies on multiple sequence alignment and provides rates of conservation known as trace scores. Trace scores in JET are calculated for each amino acid and range from 0.0, least conserved to 1.0, most conserved. We have used the iterative version of JET, iJET, which repeats the analysis n times to obtain a more reliable average score for each amino acid. After obtaining such scores, a threshold score $conserve_{th}$ is then used to designate an amino acid as conserved or not conserved. The determination of the value of this parameter and its role in narrowing the focus to the correct interaction interface while not discarding it, is detailed in Results section.

The obtained evolutionary scores can be transferred to critical points. Specifically, a critical point is given the conservation score of its closest amino acid on the molecular surface. A critical point with a conservation score greater than $conserve_{th}$ is deemed to be “active” as opposed to “passive.” The active/passive designation is inspired by work in [2]. As we detail below, active critical points are used to define surface regions of interest for alignment.

From active critical points to active triangles

We now describe how active critical points are used to build active triangles for matching during docking. For any rigid-body motion, a reference frame needs to be defined. In this work, reference frames are defined in terms of active triangles on molecular surface as follows: three critical points are used to define a triangle. At least one of these points has to be active for the triangle to be designated active, as well. A critical point p_1 with conservation score above $conserve_{th}$ is selected first. Two more critical points, p_2 and p_3 (not necessarily conserved) are then selected from the molecular surface. Their selection satisfies both angle and distance constraints. The angle constraints ensure that the points are not collinear. Points p_2 and p_3 are selected to lie no closer than 2Å and no further than 5Å from p_1. The minimum distance of 2Å ensures that two points are not on the same van der Waals (vdW) radius of an amino acid (for reference, the vdw sphere of a C_α atom is 1.94Å). The maximum distance of 5Å ensures that a triangle does not cover a lot of the molecular surface. The values for the employed angle and distance parameters are chosen from [33].

We narrow our focus to unique active triangles in order to limit the number of attempted transformations aligning geometrically-complementary active triangles and avoid redundancy. First, a lexicographic ordering of a triangle's vertices is employed to ensure that no two triangles share the first vertex in the ordering. Second, no two triangles are allowed to share their center of mass. This second constraint is ensured by hashing triangles by their center of mass. Given n critical points, ensuring satisfaction of the distance, angle, and uniqueness constraints results in fewer than n active triangles.

From active triangles to rigid-body transformation

First, one of the units, let's say A, is arbitrarily selected as the "base" unit. Therefore, the other unit B will be the "moving unit". For each unique active triangle selected from A, a matching active triangle is selected from B. The features considered for matching two active triangles are only geometric. Suppose the two selected triangles are tr_A and tr_B. The rigid-body transformation superimposing triangle tr_B over triangle tr_A according to 1, will align the monomer B on monomer A, resulting in a particular dimeric configuration. Thus, a new dimeric configuration is the result of a rigid-body transformation using active triangles.

$$T = tr'_B * tr_A \tag{1}$$

where tr'_B defines the inverse of reference frame tr_B.

Energy function

HopDock uses a simple energy function to quickly sample low-energy dimeric configurations to get a broader view of the local minima in energy minimization component. Suppose HopDock has obtained a new dimeric configuration through a rigid-body transformation that aligns two active triangles (one on each unit). We use the following simple energy function to guide the search in HopDock towards configurations that represent minima of this energy function:

$$E = E_{vdW} + E_{electrostatic} + E_{hydrogen-bonding}. \tag{2}$$

The first two terms, capturing van der Waals and electrostatic interactions, are implemented as in the CHARMM22 force field [37]. To capture the van der Waals energy we have used the standard 6-12 Lennard-Jones potential as follows:

$$E_{vdW} = \sum_{atompairs} \epsilon[(\frac{r_{ij}}{d_{ij}})^{12} - 2 \times (\frac{r_{ij}}{d_{ij}})^{6}] \tag{3}$$

where r_{ij} is the atomic radii sum, ϵ is the energy well depth derived from CHARM22 [37], and d_{ij} is the distance between atoms i and j. This energy term penalizes collisions between atoms on one unit and atoms on the other unit in the bound configuration. Atomic pairs (one atom on each unit) that lie not only closer but also farther than an ideal distance (determined by atom types) are also penalized.

The electrostatic term is computed based on Coulomb's law:

$$E_{electrostatic} = \sum_{atompairs} \frac{q_i \times q_j}{e \times {d_{ij}}^2} \tag{4}$$

where q_i and q_j are the electrostatic charges of atoms i and j obtained from CHARM22 [37], e is the dielectric constant (vacuum constant 1 is used for this paper), and d_{ij} is the distance between atoms i and j. The purpose of this term is essentially to penalize atomic pairs that bring similar charges.

The hydrogen-bonding term is calculated through the 12-10 hydrogen potential [38] as follows:

$$E_{hydrogen-bonding} = 5 \times (\frac{r_0}{d_{ij}})^{12} - 6 \times (\frac{r_0}{d_{ij}})^{10} \tag{5}$$

where d_{ij} is the distance between the interface acceptor and donor atoms i and j, and $r_0 = 2.9$ Å is the optimal distance for hydrogen bonding. The purpose of this term is to reward formation of possible hydrogen bonds between atoms in an interaction interface.

HopDock: A basin hopping algorithm to sample low-energy dimeric configurations

Main components

HopDock obtains a trajectory of n dimeric configurations $C_1, \ldots, C_n$ that correspond to minima of a chosen energy function through the BH framework. The general BH framework is illustrated in Figure 2. Starting from a configuration C_1 sampled at random (obtained through a rigid-body transformation aligning sampled geometrically-complementary active triangles), HopDock hops between two consecutive configurations in the trajectory, C_i and C_{i+1}, through an intermediate configuration $C_{\text{perturb,i}}$. A structural perturbation component in HopDock modifies C_i to obtain a configuration $C_{\text{perturb,i}}$ that allows escaping the current minimum represented by C_i. The minimization component follows the perturbation. The minimization consists of a series of modifications, starting with $C_{\text{perturb,i}}$, to reach a new configuration C_{i+1} that represents the nearest minimum to $C_{\text{perturb,i}}$. C_{i+1} is added to the trajectory according to the Metropolis criterion, based on the energetic difference between C_i and C_{i+1} and an effective temperature serving as a scaling parameter as in $e^{-[E(C_{i+1})-E(C_i)]/T_e}$. The objective is for the trajectory of energy minima to converge to lower-energy minima over time. In the Results section, we detail and analyze the effect of different temperature values to select an effective temperature that allows enhancing the sampling of low-energy minima near the native dimeric configuration.

Structural perturbation

The perturbation component modifies the current minimum C_i by seeking a new rigid-body transformation to obtain $C_{\text{perturb,i}}$. A naive implementation of the perturbation component is not careful to maintain a correlation between the contact interface in C_i and that in $C_{\text{perturb,i}}$. We compare to a naive implementation in Results section and show that preserving some of the contact interface of C_i in $C_{\text{perturb,i}}$ is important in obtaining good-quality decoy configurations. We do so by essentially limiting the search for a new contact interface in $C_{\text{perturb,i}}$ to surface regions near the contact interface in C_i. This is achieved by limiting the neighborhood over which active triangles are sought for the new transformation resulting in $C_{\text{perturb,i}}$. Studies show that applications of the BH framework are successful when the magnitude of this perturbation jump, measured through some distance function over C_i and $C_{\text{perturb,i}}$, is not too small and not too large [21, 39].

Given the current minimum C_i, a dimeric configuration $C_{\text{perturb,i}}$ that preserves some of the good structural characteristics of C_i is obtained as follows. Let us refer to the two active triangles that define the rigid-body transformation resulting in the current minimum C_i as $\{tr_A, tr_B\}$ (one from each monomer). The implementation of the perturbation component that we employ here samples a new active triangle $tr_A^{'}$

over the molecular surface of unit A uniformly at random in a d-neighborhood of tr_A. Here d refers to the distance, in angstroms, between the center of mass of tr_A and $tr^{'}_A$. Given the newly sampled $tr^{'}_A$, a new active triangle $tr^{'}_B$ is sampled uniformly at random in a d-neighborhood of tr_B. The process is repeated until a pair $tr^{'}_A$ and $tr^{'}_B$ is found that is geometrically complementary. Once these triangles are obtained, a new rigid-body transformation aligning them is defined (as described above), resulting in the perturbed dimeric configuration $C_{\text{perturb,i}}$.

Small values of d will ensure that C_i and $C_{\text{perturb,i}}$ are close in configuration space and so share structural features and parts of their contact interfaces. However, such values may result in no geometrically-complementary active triangles. Large values of d increase the probability that a geometrically-complementary pair will be sampled, but they also result in C_i and $C_{\text{perturb,i}}$ potentially being far away in configuration space. When that happens, our adaptation of BH in this algorithm degenerates to essentially minimization with random restarts. Studies show that applications of BH are successful when the magnitude of the perturbation jump, measured through some distance function over C_i and $C_{\text{perturb,i}}$, is not too small and not too large [21, 39]. In Results section we show the effect of two values of d on the magnitude of the perturbation jump and the ability of our algorithm to sample minima near the native configuration. We also show that controlling d to some not very large value yields better results than minimization with random restarts (where d is essentially infinite).

Local optimization: energy minimization

The minimization component modifies the perturbed configuration $C_{\text{perturb,i}}$ to obtain a new nearest energy minimum C_{i+1}. The minimization essentially attempts to correct the structural features that the perturbation component changed from C_i in $C_{\text{perturb,i}}$ to obtain a new set of good structural features that correspond to another energy minimum C_{i+1}. The minimization component in this paper carries out at most m consecutive structural modifications, starting with $C_{\text{perturb,i}}$ until k consecutive modifications fail to result in a lower energy. Two different implementations are pursued in this paper, depending on how the structural modifications are defined. One straightforward implementation is to define each of these modifications essentially as versions of the perturbation component, but with smaller d. The purpose of making d small is so that the minimization brings $C_{\text{perturb,i}}$ to the nearest local minimum and not to some random point in the configuration space.

Our analysis shows that it can be hard to find small values of d that will still allow finding geometrically-complementary active triangles. Therefore, this implementation is not effective, as it tends to make large jumps in configuration space as it attempts to lower the energy. The minimization is likely not to project a perturbed configuration to its nearest minimum. Therefore, a new implementation is pursued for the minimization component. This implementation essentially samples new rigid-body transformations directly, rather than through active triangles, in a continuous neighborhood of an initial transformation.

A rigid-body transformation is represented as $\langle t, u, \theta \rangle$, where t refers to the translation component, and $\langle u, \theta \rangle$ refer to the orientation component in an axis-angle representation (implemented through quaternions). In each modification in the minimization component, a new random transformation is sampled in the neighborhood of the transformation representing the configuration resulting from the previous modification. The translation and rotation components are sampled individually. A new translation component is sampled in a δ_t neighborhood of t. A new rotation component is obtained by sampling a new axis $u^{'}$ rotating around the axis u by a sampled angle value δ_ϕ; a new angle is obtained by sampling in a δ_θ neighborhood around θ.

The implementations we propose for the minimization component do not seek to identify the true basin of a local minimum. The depth of the exploration is determined by the parameter m in the minimization. Given that the decoys need to be low-energy but can be refined in detail at a later stage, this working definition of a local minimum is sufficient. For this reason, the minimization component employs a simple energy function.

Results and discussion

The organization of this section is as follows. The implementation details and the protein systems employed here for validation of HopDock are described first. Second, an analysis of the distribution of evolutionary-conserved regions on the molecular surface in finding the true interaction interface is presented in the next section. The next few sections provide a detailed analysis on how values of different parameters in HopDock have been chosen. The parameters analyzed here are the evolutionary conservation threshold, the effective temperature T_e employed in Metropolis Criterion, the perturbation distance d in the perturbation component, and the translation distance t in the minimization component. In the following section an analysis has been performed to investigate the relationship between the lower energy values to the near-native structures. A detailed comparative analysis on the attainment of the known native configuration for the proteins systems studied in this work to other state-of-the-art docking protocols is provided in the last section.

Experimental setup

Implementation details

HopDock was run on a 3GHz of Opteron Processor with 4GB of memory to generate $10,000$ dimeric configurations per protein system considered. A detailed analysis of HopDock was conducted on $5,000$ to $20,000$ configurations. Results obtained with r $10,000$ configurations were found to be representative, so the analysis presented below is over $10,000$ sampled configurations. Depending on the size of the protein systems under investigation, obtaining this number of configurations takes anywhere from 1 - 12 hours on one CPU.

Performance measurements

Our analysis employs least Root-Mean-Square-Deviation (lRMSD) to the known native dimeric structure to determine the quality of a generated configuration. lRMSD is a widely accepted performance measurements in docking methods, reported in units of Ångström (Å). RMSD is a measure of the average atomic displacement between two configurations, say x and y, under comparison and is calculated as follows:

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}\|x_i - y_i\|^2} \tag{6}$$

lRMSD refers to the minimum RMSD over all possible rigid-body motions of one configuration relative to the other. A value between 2 and 5Å is considered to be indicative of a configuration being highly similar to the known native structure, and the configuration is deemed near-native. We use lRMSD here not only to determine the proximity of dimeric configurations generated by HopDock to the known native structure but also to analyze the proximity of configurations to each other in the trajectory generated by HopDock.

Protein systems of study

We have selected seventeen different dimers with known native structures obtained from PDB as our systems of study. These dimers, listed in Table 1, are chosen because they vary in size, functional class, and have been studied by other docking methods, as well. Table 1 lists the PDB ID of the known native structure of each dimer in column 1, the size of each unit in a dimer in terms of number of atoms in column 2, and the known functional classification obtained from PDB in column 3. Systems that are CAPRI targets are marked with an asterisk in column 1.

Evolutionary conservation analysis preserves native interface

The role of evolutionary conservation in finding the true contact interface has been demonstrated in other studies [40, 41]. The work in [40] shows good correlation between sequence conservation and inclusion of conserved surface regions in the interaction interface. The analysis in [40] was expansive, classifying 265 proteins into different functional categories and measuring the correlation between conservation and inclusion of conserved regions in the interaction interface through Matthews' correlation coefficient (MCC) [42]. Analysis of MCC values allowed concluding that interaction interfaces in signal proteins and enzymes was particularly conserved. A larger dataset of 2646 protein interfaces was analyzed in [41]. The study concluded that not only there are highly conserved surface regions on the majority of proteins, but also in most proteins these regions are more likely to be found on interaction interfaces than on the rest of the molecular surface.

We conduct here a similar analysis to that in [41] to investigate the relationship between the known interaction interface and evolutionary-conserved regions based on our working definition of active critical points (described in Methods section). The goal is to uncover any correlation between high evolutionary conservation score and the true contact interface. We point out that our analysis is confined to the 17 systems studied in this work, 8 of which were specifically chosen due to their inclusion in the study in [41]. We measure two different ratios, $R_{interface}$ and R_{rest} on each of the known native dimeric structures of the systems studied here. $R_{interface}$ is the ratio between the number of conserved critical points on the known interaction interface to the total number of critical points on that interaction interface. R_{rest} is $1 - R_{interface}$. Two critical points, one from the molecular surface of unit A and the other on that of unit B in known native dimeric structure, are considered to be in contact and thus in the true interaction interface if their Euclidean distance is no higher than 5Å. This distance threshold is commonly employed in other work [2, 5].

Each of the ratios described above is measured as a percentage, and the difference between them, $R_{rest} - R_{interface}$, is plotted in Figure 3 in column diagram format. A negative value indicates that the interaction interface is more conserved than the rest of the surface, as more of the active critical points fall in it rather than elsewhere on the molecular surface. Results are shown for three different conservation thresholds in $\{0.25, 0.50, 0.75\}$ in different colors to visualize the effect that varying the conservation threshold has on the distribution of conserved critical points.

The results in Figure 3 allow concluding that in 10 of the 17 dimers, the distribution of conserved critical points is heavily concentrated on the true interaction interface as opposed to the rest of the molecular surface. In the context of employing this information for docking, this result means that more rigid-body motions will focus on matching regions on the actual interaction interface than elsewhere on the molecular surfaces. On the rest of the dimers, Figure 3 shows that the difference is not too large, which means HopDock will not spend a large portion of its time on matching regions elsewhere on the molecular surface rather than on the known interaction interface.

Analysis of different parameter values employed in HopDock

Here we investigate in detail the effect that varying values of certain parameters in HopDock has on the quality of the ensemble of sampled dimeric configurations. The parameters we investigate are the conservation threshold and its effect on the size of the search space, the effective temperature employed in the Metropolis criterion and its effect on the energetic and structural quality of sampled configurations, and the perturbation distance in the perturbation component and translation distance in the minimization component and their effect on the overall quality of the sampled ensemble.

Analysis of evolutionary conservation threshold

Analysis on the choice of the conservation threshold, $conserve_{th}$, is presented in Table 2 on three selected systems that considers the same three different thresholds as in Figure 3. The results of this table are obtained through our previous work [43, 44]. The analysis focuses on showing the effect of different evolutionary conservation thresholds on the number of active triangles and essentially on the lowest lRMSD to native that can be achieved. Column 1 indicates the PDB ID of these three systems and their chains in brackets. Three conservation thresholds, 0.25, 0.5, and 0.75 are employed to define three sets of active triangles and are reported in column 2. Column 3 shows the number of active triangles defined on the molecular surface of the base/reference unit under each conservation threshold. For each system, three sets of dimeric configurations, one for each threshold, are then obtained from our previous work. The lowest lRMSD to the known native structure from these dimeric configurations is recorded for each set and these values are reported for each system in column 4. The last column shows $R_{rest} - R_{interface}$ recorded for each conservation threshold as from Figure 3.

Table 2 highlights a few results. First, the number of active triangles goes down as the conservation threshold increases. This is expected, since a higher threshold limits the number of active critical points which in turn limit the number of active triangles. Second, the lowest lRMSD generally goes down as the conservation threshold goes up, but the effect on lRMSD depends on how much of the conserved critical points under each conservation threshold remain on the true interaction interface as opposed to elsewhere.

These results, combined with our analysis on the distribution of conserved critical points on the molecular surface above, allow us to conclude that a conservation threshold of 0.5 is a reasonable compromise between maintaining a smaller number of active triangles, thus reducing the size of the search space, while retaining the known interaction interface. For this reason, the rest of our experiments employ $conserve_{th} = 0.5$ as conservation threshold.

Analysis of effective temperature

The effective temperature T_e in the Metropolis criterion affects the probability $e^{-\delta E T_e}$ with which an energetic increase δE is accepted in the trajectory. A higher temperature increases the probability to accept an energy increase between two consecutive configurations in the trajectory than a lower temperature. A high temperature will allow HopDock to make large jumps in the energy surface, effectively degenerating to a random restart search where there is no correlation between two consecutive configurations in the HopDock trajectory. On the other hand, a low temperature may provide too strong a bias and not allow HopDock to accept temporary energetic increases to potentially cross energy barriers needed for convergence to deeper local minima over time. While we have tested various effective temperatures, the results below show the effect on sampling when using two effective temperatures, which we refer to as T_0 and T_1. T_0 is representative of a medium temperature that allows accepting an energy increase of 2 kcal/mol with probability of 0.39, whereas T_1 is representative of a lower temperature that allows accepting that same energy increase with a lower probability of 0.16.

Table 3 summarizes the effect of these two effective temperatures on the sampling of local minima by HopDock on the same 3 selected systems as the conservation threshold analysis (their PDB ids are shown in column 1). Three statistics are recorded and shown in columns 3-5. Column 3 shows the lowest lRMSD from the native structure over dimeric configurations sampled by HopDock under each effective temperature. Column 4 shows the lowest energy achieved over all HopDock-sampled configurations. Column 5 shows the percentage of configurations with lRMSD less than 5Å from the native structure.

Table 3 shows that T_0 allows HopDock to generate slightly more near-native configurations than T_1 does. As expected, T_1 achieves lower energies, but very low energies does not necessarily translate to near-native configurations. Column 3 indicates almost comparable results of T_0 and T_1 in terms of lowest lRMSD to the native structure. This analysis makes the case that the results are almost comparable for medium and low effective temperatures. Therefore, we have chosen the higher temperature T_0 to allow more energetic diversity for the rest of the experiments in the Metropolis criterion that guides the acceptance of local minima sampled by HopDock.

Analysis of perturbation distance

The next experiment compares different implementations of the perturbation distance d described in Methods section. The first implementation essentially allows testing the efficacy of random restart. In this implementation, a geometrically-complementary pair of triangles is sampled uniformly at random, and the resulting transformation is applied to obtain a new perturbed configuration. This implementation essentially refers to the case when $d = \infty$, since it employs no knowledge of the pair of triangles that were aligned to obtain the previous minimum C_i. In the second and third implementations, d is controlled and set to 7 and 5Å, respectively. A smaller value of d makes it very hard to find geometrically-complementary triangles on the molecular surfaces of the units in the dimer.

Table 4 compares these three different implementations in terms of various statistics. First, the distance in terms of lRMSD between two consecutive perturbed configurations, $C_{\mathrm{perturb,i}}$ and $C_{\mathrm{perturb,i+1}}$ is recorded to obtain a measure of the magnitude of the perturbation jump in each setting. Let us refer to this distance as l. Column 3 shows the median over these distances over all perturbed configurations obtained in the process of running HopDock with each of the three implementations of the perturbation component. This is referred to as l_m in column 3. Columns 4 and 5 provide more detail into the distribution of these distances by showing the percentage of distances in the $0-5$Å and $5-10$Å range, respectively. Finally, columns 6 and 7 show the effect of controlling d on the ensemble of sampled minima through the lowest lRMSD to the native structure and the percentage of minima configurations with lRMSD to the native structure less than 5Å.

The results shown in Table 4 allow drawing a few conclusions. First, controlling d allows reducing the median distance l_m. The number of consecutive perturbed configurations with < 5Å and < 10Å lRMSD of each-other also increases with lower d. In addition, the number of minima with low lRMSDs to the native structure is not impacted negatively by lowering d. For this reason, the rest of our experiments employ the implementation of the perturbation component with $d = 5$Å.

Analysis of minimization translation distance

The third experiment pitches different implementations of the minimization component against one another. In all implementations, $m = 100$ and $k = 20$ (i.e., the minimization trajectory initiated from a perturbed configuration is at most 100 steps long and can terminate earlier, if 20 consecutive steps all fail to lower energy). In the Methods section we explain that the consecutive modifications pursued in the minimization component to lower interaction energy can be considered versions of the perturbation component with $d = 5$Å. However, these are large moves and do not ensure that the minimization will project a perturbed configuration to its nearest local minimum. Our analysis suggests that this implementation is less effective. The results below show the effect of three other implementations, which sample rigid-body transformations directly, rather than through active triangles, in a continuous neighborhood of the rigid-body transformation in the previous configuration in the minimization trajectory. These implementations use the same thresholds of $\delta_\phi = 10°$ and $\delta_\theta = 30°$. What varies is the translation distance threshold t, which takes values in $\{1.5, 2.0, 2.5\}$Å. We decide to focus on varying t rather than the orientation, as the translation distance is expected to have a more dramatic effect on changing the contact interface. The goal of this analysis is to determine whether making small moves during minimization, which increases the probability of actually populating the minimum nearest to $C_{\text{perturb,i}}$, has any effect on the distance between a perturbed configurations and its nearby minimum as well as on the overall quality of sampled local minima.

Table 5 compares these implementations in terms of various statistics. First, the distance between $C_{\text{perturb,i}}$ to C_i sampled by HopDock is recorded; let's refer to this distance as i. Column 3 shows the median of these distances over the trajectory of minima obtained by HopDock, referred to as i_m. Columns 4 and 5 show the percentage of distances i in the $0 - 5$Å and $5 - 10$Å range, respectively. Finally, columns 6 and 7 show the effect of i on the ensemble of sampled minima through the lowest lRMSD to the native structure and the percentage of minima configurations with lRMSD to the native structure less than 5Å.

The results shown in Table 5 suggest that varying t in this range does not seem to result significant differences in the measured statistics. However, comparing the lowest lRMSD to the native structure and the overall number of minima within 5Å of the native structure shows that a translation distance $t = 1.5$Å provides a good compromise.

Analysis of relationship between energy and proximity to native structure

Based on the above analysis, an ensemble of $10,000$ dimeric configurations is obtained for each of the 17 protein systems using HopDock with $conserve_{th} = 0.5$, $T_e = T_0$, $d = 5$Å , and $t = 1.5$Å. Here we investigate the extent to which an energetic reduction scheme that reduces the sampled ensemble Ω based on low energies is able to retain near-native configurations with low lRMSDs to the known native structure. For this purpose, we define a variable p to track the % of configurations with lowest energies retained in a reduced ensemble Ω_p. p is varied from $10 - 100$% increments of 10. $\Omega_{p=100}$ means that the entire ensemble Ω is retained.

$\Omega_{p=10}$ means that only the 10% of the configurations with lowest energies are retained. Figure 4 plots the lowest lRMSD to the native structure over configurations in each reduced ensemble Ω_p as p is varied. This is shown for each of the 17 protein systems. Figure 4 shows that the lowest lRMSD obtained by HopDock over the entire ensemble is retained even when $p = 40 - 50\%$ for most protein systems. This effectively means that near-native configurations are not discarded if focusing only on those with energies below the mean. For many systems, the lowest lRMSD does not change significantly even when only p is further reduced. This result is encouraging, particularly, if HopDock is considered as a configuration sampling technique to be employed in docking protocols. Even this coarse reduction by energy ensures that near-native configurations will be present in the reduced ensemble and can be further refined by docking protocols to improve their proximity to the native structure.

Comparative analysis of HopDock to other existing methods

Table 6 shows the results obtained when applying HopDock (with the parameter values and implementations above) on the seventeen protein systems studied here. Columns 1-2 relate details on these dimers in terms of the PDB ID of the known native structure of the dimer and size (total number of atoms). Column 3 shows the lowest lRMSDs to the native structure reported by work in Budda [9], which uses geometric complementarity to match regions for docking and relies on clustering and some short energetic refinements of top clusters. Column 4 shows again for reference the lowest lRMSD obtained from our previous work which relies on exhaustive matching of active geometrically-complementary triangles [44]. Columns 5 and 6 show the lowest lRMSDs obtained on each dimer with the pyDock [4] and ClusPro [6] servers, respectively. The methods selected for the comparison are representative of geometry- and energy-based methods commonly used by docking protocols. Column 7 reports the lowest lRMSD obtained by the HopDock (the lowest-lRMSD configuration is shown superimposed on the native structure for 9 systems in Figure 5).

Table 6 shows that HopDock achieves low lRMSDs to the native structure on each system. These lRMSDs are comparable to the geometric-based Budda method and our previous work in [44] on most protein systems. In few cases, our previous work, which relies on exhaustive sampling of geometrically-complementary active triangles, achieves slightly lower lowest lRMSDs than HopDock. This is largely due to the fact that a very large number of dimeric configurations, $100,000$ to $700,000$, depending on the size of the proteins are sampled in [44]. It is indeed encouraging to obtain similar lRMSDs, and even lower in some cases, with HopDock, which samples only about 10% of the configurations in [44]. The comparison of HopDock to two energy-based methods, pyDock and ClusPro, allows us to obtain a more comprehensive view of the results obtained by HopDock. In comparison to PyDock and ClusPro, the performance of HopDcok in terms of proximity to the known native structure is better or comparable. This is particularly the case as the size of the protein system grows. This result is expected, as energy-based optimization methods operate on a large search space, whereas HopDock focuses on potentially-relevant contact interfaces through its combination of geometry and evolutionary conservation.

Comparative analysis of computing time and power

The above results are promising and suggest that HopDock is an important first step into a multi-stage docking protocol. Here we provide a broader picture by comparing HopDock to two established docking servers, pyDock [4] and ClusPro [6]. The purpose of this analysis is to better gage how HopDock compares to these servers, even though we are fully aware that the implementations in these servers are tuned and optimized, and that the methods in these servers use different search approaches and sophisticated energy functions. To get a sense of the resources available to HopDock as compared to these other serves, Table 7 summarizes number of processors, processing speed, memory, and average CPU time used by each method. Table 8 shows the time that each of these servers, including our own algorithm, HopDock, takes on each of the 17 systems studied here. This table shows that ClusPro is faster on most systems, but HopDock, though not fine tuned, achieves comparable running times to both servers. Taken together, these results suggest that HopDock is a promising search algorithm that ca be used in the first stage by docking protocols.

Conclusion

We have presented a novel probabilistic search algorithm, HopDock, to efficiently generate low-energy decoys for protein-protein docking. The algorithm conducts its search over rigid-body transformations that align evolutionary conserved geometrically-complementary regions on molecular surfaces. The incorporation of evolutionary information reduces and simplifies the search space over which the structural perturbation in HopDock selects the dimeric configurations for minimization component. Since HopDock is a decoy sampling algorithm, a simple energy function has been employed to reduce the time complexity of the minimization component and measure the performance of HopDock.

A detailed analysis of evolutionary conservation and different components of HopDock has been presented in this work. This analysis shows that HopDock produces many near-native configurations in the decoy ensemble it samples. A detailed comparative analysis shows that the algorithm is competitive with other state-of-the-art protocols. This suggests that HopDock is a promising decoy sampling algorithm to be incorporated in a docking protocols.

There are several further directions for future research. One involves taking into account additional criteria beyond evolutionary conservation to predict interaction interfaces [45, 46]. Pursuit of more sophisticated energy functions used by state-of-the-art docking protocols is another direction. We also intend to pursue different implementations, especially for the minimization component. While the perturbation component can focus on obtaining low-resolution configurations over the SE(3) space of rigid-body transformations, the minimization can add more detail to project a coarse-grained configuration onto a nearby accurate minimum. Combined with clustering and further refinement of top-populated clusters, the combination of a geometric and energetic treatment proposed here promises to result in an effective docking protocol. Wider sampling through population-based versions of the BH framework and incorporation of fluctuations on and nearby interaction interfaces will also be considered to improve the quality of decoy ensembles.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

IH suggested the methods and the performance study in this manuscript and drafted the manuscript. AS guided the study, provided comments and suggestions on the methods and performance evaluation, and improved the manuscript writing.

Acknowledgements

This work is supported in part by NSF IIS CAREER Award No. 1144106.

References

1. Vajda S, Kozakov D: **Convergence and combination of methods in protein-protein docking** 2009, **19**:164–170.
2. Dominguez C, Boelens R, Bonvin A: **HADDOCK: A protein-protein docking approach based on biochemical orbiophysical information** 2003, **125**:1731–1737.
3. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE: **The Protein Data Bank**. *Nucleic Acids Research* 2000, **28**:235–242, [www.pdb.org].
4. Cheng TMK, Blundell TL, Fernandez-Recio J: **pyDock: Electrostatics and desolvation for effective scoring of rigid-body protein–protein docking**. *Proteins* 2007, **68**(2):503–515.
5. Chen R, Li L, Weng Z: **ZDock: an initial-stage protein-docking algorithm** 2003, **52**:80–87.
6. Comeau SR, Gatchell DW, Vajda S, Camacho CJ: **ClusPro: a fully automated algorithm for protein-protein docking** 2004, **32**(S1).
7. Schneidman-Duhovny D, Inbar Y, Nussinov R, Wolfson HJ: **PatchDock and SymmDock: servers for rigid and symmetric docking** 2005, **33**(S2):W363–W367.
8. Inbar Y, Benyamini H, Nussinov R, Wolfson HJ: **Combinatorial docking approach for structure prediction of large proteins and multi-molecular assemblies** 2005, **2**:S156–S165.
9. Polak V: **Budda: backbone unbound docking application**. *Master's thesis*, School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel 2003.
10. Lyskov S, Gray JJ: **The RosettaDock server for local protein-protein docking** 2008, **36**(S2):W233–W238.
11. Terashi G, Takeda-Shitaka M, Kanou K, Iwadate M, Takaya D, Umeyama H: **The SKE-DOCK server and human teams based on a combined method of shape complementarity and free energy estimation** 2007, **69**(4):866–887.
12. Mashiach E, Nussinov R, Wolfson HJ: **FiberDock: Flexible induced-fit backbone refinement in molecular docking** 2010, **78**(6):1503–1519.
13. Lensink MF, Wodak SJ: **Blind predictions of protein interfaces by docking calculations in CAPRI** 2010, **78**(15):3085–3095.
14. Gray JJ: **High-resolution protein-protein docking** 2006, **16**(2):183–193.
15. Moitessier N, Englebienne P, Lee D, Lawandi J, Corbeil C: **Towards the development of universal, fast and highly accurate docking/scoring methods: a long way to go**. *British J Pharmacology* 2009, **153**(S1):S7–S27.
16. Vajda S, Kozakov D: **Convergence and combination of methods in protein–protein docking**. *Current opinion in structural biology* 2009, **19**(2):164–170.
17. David J, Doye JP: **Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms** 1997, **101**(28):5111–5116.
18. Verma A, Schug A, Lee K, Wenzel W: **Basin hopping simulations for all-atom protein folding** 2006, **124**(4):044515.
19. Prentiss MC, Wales DJ, Wolynes PG: **Protein structure prediction using basin-hopping.** *The Journal of Chemical Physics* 2008, **128**(22):225106–225106.
20. Olson B, Shehu A: **Populating Local Minima in the Protein Conformational Space** 2011:114–117.
21. Olson B, Shehu A: **Evolutionary-inspired Probabilistic Search for Enhancing Sampling of Local Minima in the Protein Energy Surface**. *Proteome Sci* 2012. in press.
22. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E: *The journal of chemical physics", journal = "J. Chem. Phys.", number = "6", pages = "1087–1092", title = "Equation of State Calculations by Fast Computing Machines", volume = "21", year = "1953"*.
23. Akbal-Delibas B, Hashmi I, Shehu A, Haspel N: **Refinement of Docked Protein Complex Structures Using Evolutionary Traces**. In *Comput Struct Biol Workshop* 2011:400–404.
24. Gabb HA, Jackson RM, Sternberg MJ, et al.: **Modelling protein docking using shape complementarity, electrostatics and biochemical information**. *Journal of molecular biology* 1997, **272**:106–120.

25. Mandell JG, Roberts VA, Pique ME, Kotlovyi V, Mitchell JC, Nelson E, Tsigelny I, Ten Eyck LF: **Protein docking using continuum electrostatics and geometric fit**. *Protein Eng.* 2001, **14**(2):105–113.

26. Murphy J, Gatchell DW, Prasad JC, Vajda S: **Combination of scoring functions improves discrimination in protein-protein docking** 2003, **53**(4):840–854.

27. Venkatraman V, Yang YD, Sael L, Kihara D: **Protein-protein docking using region-based 3D Zernike descriptors**. *BMC bioinformatics* 2009, **10**:407+.

28. Connolly ML: **Analytical Molecular Surface Calculation**. *J. Appl. Cryst.* 1983, **16**(5):548–558.

29. Lin SL, Nussinov R, Fischer D, Wolfson HJ: **Molecular surface representations by sparse critical points.** *Proteins* 1994, **18**:94–101.

30. Wolfson HJ, Rigoutsos I: **Geometric hashing: an overview**. *IEEE Comp Sci and Engineering* 1997, **4**(4):10–21.

31. Esquivel-Rodríguez J, Yang YD, Kihara D: **Multi-LZerD: Multiple protein docking for asymmetric complexes.** *Proteins* 2012.

32. Chen BY, Honig B: **VASP: a volumetric analysis of surface properties yields insights into protein-ligand binding specificity**. *PLoS Comput Biol* 2010, **6**(8).

33. Fischer D, Lin SL, Wolfson HL, Nussinov R: **A geometry-based suite of molecular docking processes** 2005, **248**(2):459–477.

34. Lichtarge O, Bourne HR, Cohen FE, et al.: **An Evolutionary Trace Method Defines Binding Surfaces Common to Protein Families** 1996, **257**(2):342–58.

35. Engelen S, Trojan LA, Sacquin-Mora S, Lavery R, Carbone A: **A Large-Scale Method to Predict Protein Interfaces Based on Sequence Sampling**. *PLoS Comp Bio* 2009, **5**:e1000267.

36. Goldenberg O, Erez E, Nimrod G, Ben-Tal N: **The ConSurf-DB: pre-calculated evolutionary conservation profiles of protein structures**. *Nucleic Acids Research* 2009, **37**:D323–D327.

37. Brooks BR, Bruccoleri RE, Olafson BD, Swaminathan S, Karplus M, et al.: **CHARMM: A program for macromolecular energy, minimization, and dynamics calculations**. *J. Comput. Chem.* 1983, **4**(2):187–217.

38. Kortemme T, Baker D: **A simple physical model for binding energy hot spots in protein–protein complexes**. *Proc. Natl Acad of Sci USA* 2002, **99**(22):14116–14121.

39. Lourenço H, Martin O, Stützle T: **Iterated Local Search**. In *Handbook of Metaheuristics, Volume 57 of* Operations Research & Management Science. Edited by Glover F, Kochenberger G, Kluwer Academic Publishers 2002:321–353.

40. Kanamori E, Murakami Y, Tsuchiya Y, Standley DM, Nakamura H, Kinoshita K: **Docking of protein molecular surfaces with evolutionary trace analysis**. *proteinssfb* 2007, **69**:832–838.

41. Choi YS, Yang JS, Choi Y, Ryu SH, Kim S: **Evolutionary conservation in multiple faces of protein interaction**. *Proteins* 2009, **77**:14–25.

42. Matthews BW, et al.: **Comparison of the predicted and observed secondary structure of T4 phage lysozyme**. *Biochimica et Biophysica Acta (BBA) - Protein Structure* 1975, **405**:442–451.

43. Hashmi I, Akbal-Delibas B, Haspel N, Shehu A: **Protein Docking with Information on Evolutionary Conserved Interfaces**. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2011 IEEE International Conference on* 2011:358–365.

44. Hashmi I, Akbal-Delibas B, Haspel N, Shehu A: **Guiding Protein Docking with Geometric and Evolutionary Information** 2012, **10**(3):1242008.

45. Glaser F, Steinberg DM, Vakser IA, Ben-Tal N: **Residue frequencies and pairing preferences at protein-protein interfaces.** *Proteins* 2001, **43**(2):89–102.

46. Chen CT, Peng HP, Jian JW, Tsai KC, Chang JY, Yang EW, Chen JB, Ho SY, Hsu WL, Yang AS: **Protein-Protein Interaction Site Predictions with Three-Dimensional Probability Distributions of Interacting Atoms on Protein Surfaces**. *PLoS ONE* 2012, **7**(6):e37706+.

47. Humphrey W, Dalke A, Schulten K, et al.: **VMD – Visual Molecular Dynamics**. *Journal of Molecular Graphics* 1996, **14**:33–38.

Figures

Figure 1 - Protein-protein Docking

Two unbound units A and B are docked to form a bound configuration AB through rigid-body motion.

Figure 2 - Overview of Basin Hopping Framework

Under the BH framework, the energy surface is transformed into a collection of interpenetrating staircases. A trajectory of local minima is obtained consecutively, through iterated applications of a structural perturbation to jump out of a current local minimum and an ensuing local optimization to map to another nearby local minimum.

Figure 3 - Analysis of Evolutionary Conservation

Column diagram shows the $R_{rest} - R_{interface}$ difference on each of the seventeen dimers. Three different conservation thresholds are considered and results for each are shown in different colors. Negative percentages indicate the interface is more conserved than the rest of the molecular surface.

Figure 4 - Analysis of Top Lowest Energy Values

Lowest lRMSD from native structure is shown for each reduced Ω_p ensemble on all 17 protein systems. Ω_p contains the p% lowest-energy configurations generated by HopDock. Results for each protein system are shown in different colors.

Figure 5 - Lowest lRMSD to Native Structures

Nine systems are selected to draw the lowest-lRMSD configuration obtained by HopDock. This configuration is drawn in opaque, with chains in blue and red. The native structure, over which the lowest-lRMSD configuration is superimposed to highlight structural differences, is drawn in transparent. The actual lRMSD between the two is shown below. Visualization is obtained through VMD [47].

Tables

Table 1 - Protein systems of study

Details are provided on the dimers selected in this study. PDB ID of known native structure along with chains in bracket is shown in column 1. Column 2 lists the number of atoms per unit in a dimer, and column 3 shows the known functional classification of each dimer. CAPRI targets are marked with an asterisk.

PDB ID (Chains)	Size(Number of Atoms)	Functional Classification
1C1Y (A,B)	1376, 658	Signaling Protein
1DS6 (A,B)	1413, 1426	Signaling Protein
1TX4 (A,B)	1579, 1378	Complex(gtpase Activatn/proto Oncogene)
1WWW (W,Y)	862, 782	Nerve Growth Factor/trka Complex
1FLT (V,Y)	770, 758	Complex (growth Factor/transferase)
1IKN (A,C)	2262, 916	Transcription Factor
1IKN (C,D)	916, 1589	Transcription Factor
1VCB (A,B)	755, 692	Transcription
1VCB (B,C)	692, 1154	Transcription
1OHZ* (A,B)	1027, 416	Cell Adhesion
1T6G* (A,C)	2628, 1394	Hydrolase Inhibitor
1ZHI* (A,B)	1597, 1036	Transcription/replication
2HQS* (A,C)	3127, 856	Transport Protein/lipoprotein
1QAV (A,B)	663, 840	Membrane Protein/oxidoreductase
1G4Y (B,R)	682, 1156	Signaling Protein
1CSE (E,I)	1920, 522	Complex(serine Proteinase Inhibitor)
1G4U (R,S)	1398, 2790	Signaling Protein

Table 2 - Conservation threshold analysis

Table shows the effect of three different conservation thresholds 0.25, 0.5, and 0.75 on the number of active triangles and lowest lRMSD to native structure.

PDB ID	Threshold	Nr. Triangles	lRMSD (Å)	$R_{rest} - R_{interface}$
1FLT (V,Y)	0.25	2417	2.06	6.67
	0.50	2338	1.12	-3.50
	0.75	2080	1.03	1.27
1WWW (W, Y)	0.25	2900	2.29	-18.37
	0.50	2911	2.24	-7.45
	0.75	2854	2.60	6.62
1C1Y (A,B)	0.25	3385	1.89	-13.65
	0.50	3325	1.30	-38.64
	0.75	3306	1.45	-21.01

Table 3 - Effective temperature on representative systems

The effect of two effective temperatures in Metropolis Criterion is shown on three selected systems. Three different statistics are presented for each temperature value: lowest lRMSD to the native structure, lowest energy, and percentage of configurations with lRMSD to native less than 5Å.

PDB ID	T_e	lRMSD (Å)	Energy (kcal/mol)	$< 5\mathring{A}$ (%)
1FLT	T_0	1.47	-1.73	0.21
	T_1	2.10	-0.67	0.09
1WWW	T_0	2.22	-0.88	0.14
	T_1	2.16	-21.39	0.12
1C1Y	T_0	1.95	-0.83	0.78
	T_1	1.42	-10.50	0.80

Table 4 - Effect of perturbation distance on representative systems

The effect of perturbation distance on the same three representative systems is shown here. Three different values of d are considered, ∞, 5Å and 7Å. Columns 3-5 show statistics on the l distribution of lRMSDs between two consecutive perturbed configurations. Column 3 shows median lRMSD, l_m, whereas columns 4-5 show % of cases where $l \leq 5$ and 10Å, respectively. Column 6 shows lowest lRMSD over entire HopDock ensemble to native structure, and column 7 shows % of sampled configurations within 5Åof the native structure.

PDB ID	d (Å)	$l_{\rm m}$ (Å)	l<5Å(%)	l<10Å(%)	lRMSD(Å)	lRMSD<5Å(%)
1FLT	$d = \infty$	16.73	0.29	3.82	3.37	0.11
	$d = 7$	14.76	2.09	16.57	2.48	0.12
	$d = 5$	13.48	4.73	18.20	1.47	0.23
1WWW	$d = \infty$	17.83	0.27	2.86	2.33	0.12
	$d = 7$	14.62	2.19	13.32	3.63	0.08
	$d = 5$	14.22	4.86	14.91	2.22	0.08
1C1Y	$d = \infty$	14.31	0.51	9.87	2.17	0.70
	$d = 7$	11.23	4.86	30.02	2.09	0.71
	$d = 5$	11.06	6.49	31.00	1.95	0.86

Table 5 - Effect of translation distance on representative systems

The effect of translation distance t is shown on the three representative systems for t = 1.5, 2.0, and 2.5. Columns 3-5 show statistics on the i distribution of lRMSDs between the perturbed configuration and the nearby local minimum where the minimization projects the perturbed configuration. Column 3 shows the median value i_m, whereas columns $4-5$ show % of cases where $i \leq 5$ and 10 Å, respectively. Column 6 shows lowest lRMSD over entire HopDock ensemble to native structure, and 7 shows % of sampled configurations within 5Å of the native structure.

PDB ID	t (Å)	i_{m} (Å)	i<5Å(%)	i<10Å(%)	lRMSD (Å)	lRMSD<5Å(%)
1FLT	1.5	10.14	19.30	29.69	1.47	0.07
	2.0	9.81	20.56	30.66	1.90	0.15
	2.5	9.98	20.26	29.82	1.69	0.23
1WWW	1.5	9.74	21.12	30.30	2.22	0.21
	2.0	9.68	21.50	30.55	3.88	0.12
	2.5	9.50	21.85	31.43	2.98	0.08
1C1Y	1.5	9.04	24.41	31.94	1.95	0.92
	2.0	9.40	21.99	31.78	2.14	0.51
	2.5	9.56	21.62	31.71	1.05	0.86

Table 6 - Final Results of HopDock presented in this work

A comparative analysis has been performed on all seventeen dimers. For the comparison, two geometry-based (Budda [9] and work in [44]) and two energy-based methods (pyDock [4] and ClusPro [6]) have been chosen. Lowest lRMSD to native structure is reported for Budda in column 3, work in [44] in column 4, pyDock in column 5, ClusPro in column 6, and HopDock in column 7.

PDB ID (Chains)	Size	Budda [9] (Å)	Prev. [44] (Å)	pyDock [4] (Å)	ClusPro [6] (Å)	HopDock (Å)
1C1Y (A,B)	2034	1.2	1.3	10.4	7.2	1.9
1DS6 (A,B)	2839	1.2	1.8	0.8	1.7	3.4
1TX4 (A,B)	2957	1.4	2.4	18.5	4.7	1.0
1WWW (W,Y)	1644	11.4	2.2	18.2	17.2	2.2
1FLT (V,Y)	1528	1.5	1.1	2.8	4.7	1.5
1IKN (A,C)	3178	1.2	2.0	20.1	19.7	2.2
1IKN (C,D)	2505	2.0	2.0	16.7	20.9	4.6
1VCB (A,B)	1447	0.7	2.1	1.4	1.9	3.6
1VCB (B,C)	1846	1.3	1.3	22.7	1.9	1.7
1OHZ (A,B)	1443	1.8	1.7	7.5	3.3	2.2
1T6G (A,C)	4022	1.6	2.5	0.1	14.8	2.5
1ZHI (A,B)	2633	25.3	1.7	23.8	24.1	3.3
2HQS (A,C)	3983	29.1	2.2	15.2	16.6	2.6
1QAV (A,B)	1503	1.4	1.0	9.6	1.7	2.6
1G4Y (B,R)	1838	0.8	2.3	26.2	1.9	4.1
1CSE (E,I)	2442	0.7	1.5	13.2	1.1	2.7
1G4U (R,S)	4188	1.0	2.2	27.6	16.1	5.6

Table 7 - Summary of the computing power of different docking protocols

Summary of computing power of different protein-protein docking protocols is presented to place HopDock in context.

Protocols	Number of Processor	Processing Speed (GHz)	Memory(GB)	Average CPU Time (Hours)
pyDock	2 Nodes(16 core each)	2.4	65	3-5
ClusPro	16	1.3	32 (shared)	4
HopDock	2 (Core 2 Duo)	3.00	4	3-5

Table 8 - Timing comparison

Timing comparison (HH:MM) of HopDock to other web servers is presented on each of the seventeen systems studied here.

PDB ID (Chains)	HopDock (HH:MM)	pyDock [4] (HH:MM)	ClusPro [6] (HH:MM)
1C1Y (A,B)	04:00	01:30	00:53
1DS6 (A,B)	06:26	02:00	01.30
1TX4 (A,B)	10:42	02:30	01:00
1WWW (W,Y)	03:12	01:00	00:53
1FLT (V,Y)	02:36	00:30	00:53
1IKN (A,C)	06:00	01:30	01.24
1IKN (C,D)	03:54	00:18	01.24
1VCB (A,B)	01:04	00:33	00.54
1VCB (B,C)	01:36	01:08	00:58
1OHZ (A,B)	00:57	02:30	01:00
1T6G (A,C)	11:04	04:00	00:59
1ZHI (A,B)	03:17	04:45	00:59
2HQS (A,C)	12:07	01:00	01:00
1QAV (A,B)	01:05	01:30	00:40
1G4Y (B,R)	03:13	05:29	00:59
1CSE (E,I)	02:48	02:00	00:38
1G4U (R,S)	09:19	01:14	01:26